In The Claims

Kindly enter the claim amendments, without prejudice, as set forth below. A complete listing of the claims is provided, with a parenthetical indication of the status of each claim, and markings to show current changes.

1. (currently amended) A method of ~~configuring an operating system to~~ merging interrupt streams of a first type and a second type into a single service routine upon receipt of the interrupt streams by a processor ~~of the type~~ having operationally sequential placement of first type and second type vector addresses~~handle multiple interrupt types~~, said method comprising the steps of:

~~(a)providing an interrupt vector table having a first vector address executable upon receipt of an interrupt request of a first type, and a second vector address executable upon receipt of an interrupt request of a second type, the first vector address preceding the second vector address in the vector table;~~

(a)~~(b)~~providing a common interrupt dispatcher;

~~(c)~~(b) inserting an instruction ~~into~~at the first type vector address that disables interrupts of the second type~~interrupt mode~~;

~~(d)~~(c) inserting at the second type vector address, an other instruction that branches to the common interrupt dispatcher;

~~(e)~~(d) providing the common interrupt dispatcher with an interrupt routine that processes an~~the~~ interrupt, and then re-enables interrupts of the second type~~interrupt modes;~~

~~(f)wherein interrupt requests are processed by the common interrupt dispatcher without interruption.~~

2. (currently amended) The method of claim 1, wherein said inserting step (b)~~(c)~~ comprises inserting a single instruction ~~into~~at the first vector address that disables the first type and the second type interrupts ~~mode and the second interrupt mode~~.

3. (currently amended) The method of claim 2, wherein said providing step (d)~~(c)~~ comprises re-enabling the first type and second type interrupt ~~modes~~.

4. (currently amended) The method of claim 1, wherein said providing step (d)~~(c)~~ further comprises determining whether a received interrupt was of the first ~~interrupt~~ type or of the second ~~interrupt~~ type.

5. (currently amended) The method of claim 4, wherein the providing step (d)~~(c)~~ comprises checking a~~the~~ mode identifier to identify the type of interrupt.

6. (currently amended) The method of claim 4, wherein said providing step (d)~~(c)~~ further comprises the steps of providing a first interrupt handler and a second interrupt handler, and providing the interrupt routine with an instruction that

branches first type interrupts to the first interrupt handler, and branches second type interrupts to the second interrupt handler.

7. (currently amended) The method of claim 1, wherein the first type interrupt is an IRQ interrupt, and the second type interrupt is an FIQ interrupt.

8. (original) The method of claim 1, wherein the instruction is a single instruction.

9. (currently amended) A method of configuring an operating system of an ARM® processor ~~to~~ having operationally sequential IRQ and FIQ vector addresses, to ~~handle~~ IRQ and FIQ interrupt streams~~interrupts~~, said method comprising the steps of:

(a) providing a common interrupt dispatcher;

(b) inserting ~~into~~at the~~an~~ IRQ vector address ~~of an interrupt vector table~~, a single~~an~~ instruction that disables an FIQ interrupt mode;

(c) inserting ~~into~~at the~~an~~ FIQ vector address ~~of the interrupt vector table~~, an other instruction that branches to the common interrupt dispatcher; and

(d) providing the common interrupt dispatcher with an interrupt routine that processes an interrupt, and then re-enables the FIQ interrupt mode.

~~(d);~~

~~(e)wherein IRQ and FIQ interrupts are processed by the common interrupt dispatcher without interruption~~.

10. (currently amended) The method of claim 9, wherein said inserting step (b) comprises inserting an instruction ~~into~~at the IRQ vector address that disables an~~the~~ IRQ interrupt mode and the FIQ interrupt mode.

11. (original) The method of claim 10, wherein said providing step (d) comprises re-enabling the IRQ and FIQ interrupt modes.

12. (original) The method of claim 9, wherein said providing step (d) further comprises determining whether a received interrupt was an IRQ interrupt or an FIQ interrupt.

13. (currently amended) The method of claim 12, wherein said providing step (d) comprises checking a~~the~~ mode status indicator to identify the type of interrupt.

14. (currently amended) The method of claim 12, wherein said providing step (d) further comprises the steps of providing an IRQ interrupt handler and an FIQ interrupt handler, and providing the interrupt routine with an instruction that branches IRQ interrupts to the IRQ~~first~~ interrupt handler, and branches FIQ interrupts to the FIQ interrupt handler.

15. (currently amended) A method of operating a processor capable of receiving ~~having~~ multiple interrupt types, at operationally sequentially placed~~ment of~~ first type and second type interrupt vector addresses, said method comprising the steps of:

(a) providing a common interrupt dispatcher having an interrupt routine that checks a mode identifier to determine whether a received interrupt was of a first or second type, and processes the interrupt;

(b) inserting ~~into a~~at the first type vector address ~~of an interrupt vector table executable upon receipt of an interrupt of the first type~~, an instruction that disables subsequent interrupts of the first and second types;

(c) inserting at ~~a~~the second type vector address ~~of the interrupt vector table executable upon receipt of an interrupt of the second type~~, an other instruction that branches to the common interrupt dispatcher;

(d) receiving an interrupt of the first type

(e) at the first type vector address;

(f) setting the mode identifier to indicate an interrupt of the first type was received;

(g) executing the instruction to disable the first and second interrupt types;

(h) executing the other instruction to branch to the common interrupt dispatcher;

(i) processing the interrupt of the first type with the common interrupt dispatcher without interruption; and

(j) re-enabling the first and second interrupt types.

16. (original) The method of claim 15, wherein the processor is an ARM® processor and the first interrupt type is an IRQ interrupt and the second interrupt type is an FIQ interrupt.

17. (currently amended) A method of operating a processor capable of receiving~~having~~ multiple interrupt types, at operationally sequentially placed~~ment~~ first type and second type interrupt vector addresses, said method comprising the steps of:

(a) providing a common interrupt dispatcher having an interrupt routine that checks a mode identifier to determine whether a received interrupt was of a first or second interrupt type, and processes interrupts of both the first and second interrupt types;

(b) inserting at ~~a~~the second vector address ~~of an interrupt vector table executable upon receipt of an interrupt of the second type~~, an instruction that branches to the common interrupt dispatcher;

(c) receiving an interrupt of the second type

(d) at the second vector address;

(e) setting the mode identifier to indicate an interrupt of the second type was received;

(f) executing the instruction to branch to the common interrupt dispatcher; and

(g) processing the interrupt of the second type with the common interrupt dispatcher without interruption.

18. (original) The method of claim 17, wherein the processor is an ARM® processor, the first interrupt type is an IRQ interrupt and the second interrupt type is an FIQ interrupt.

19. (currently amended) An operating system for a processor having a facility to handle ~~multiple interrupt types,~~ interrupt streams of a first type and a second type upon receipt of the interrupt streams at operationally sequentially placed first type and second type vector addresses, said operating system comprising:

(a) an instruction that disables first and second interrupt types, disposed in ~~a~~the first type vector address ~~of an interrupt vector table executable upon receipt of an interrupt of a first type~~;

(b) an other instruction that branches to a common interrupt dispatcher, disposed at ~~a~~the second type vector address ~~of the interrupt vector table executable upon receipt of an interrupt of a second type~~;

(c) the common interrupt dispatcher having an interrupt routine that checks a mode identifier to determine whether a received interrupt was of the first or second type, processes the interrupt, and

then re-enables the first and second interrupt types.

20. (original) The operating system of claim 19, wherein interrupts of the first and second types are processed by the common interrupt dispatcher without interruption.

21. (currently amended) The operating system of claim 19, wherein the processor is~~adapted for use with~~ an ARM® processor, ~~wherein~~ the first interrupt type is an IRQ interrupt, and the second interrupt type is an FIQ interrupt.

22. (currently amended) A method of operating an ARM® processor, comprising the steps of:

(a) providing the system of claim 21;

(b) receiving an IRQ interrupt;

(c) branching to an IRQ vector address;

(d) setting a~~the~~ mode identifier to indicate an IRQ interrupt was received;

(e) executing the ~~single~~ instruction to disable the IRQ and FIQ interrupts;

(f) executing the other instruction to branch to the common interrupt dispatcher;

(g) processing the IRQ interrupt with the common interrupt dispatcher without interruption; and

(h) re-enabling the IRQ and FIQ interrupts.

23. (currently amended) A method of operating an ARM® processor, comprising the steps of:

(a) providing the system of claim 21;

(b) receiving an FIQ interrupt;

(c) branching to an FIQ vector address;

(d) setting the mode identifier to indicate an FIQ interrupt was received;

(e) executing the instruction at the FIQ vector to branch to the common interrupt dispatcher;

(f) processing the FIQ interrupt with the common interrupt dispatcher without interruption.

24. (currently amended) An operating system kernel for a processor having a facility to handle ~~multiple~~ interrupt streams of a first type and a second type~~s,~~ upon receipt of the interrupt streams at operationally sequentially placed first type and second type vector addresses, said operating system kernel comprising:

(a) an instruction that disables the first and second interrupt types, disposed in ~~a~~the first type vector address ~~of an interrupt vector table executable upon receipt of an interrupt of a first type~~;

(b) an other instruction that branches to a common interrupt dispatcher, disposed at ~~a~~the second vector address ~~of the interrupt vector table executable upon receipt of an interrupt of a second type~~;

(c) the common interrupt dispatcher having an interrupt routine that checks a mode identifier to

determine whether a received interrupt was of the first or second type, processes the interrupt, and then re-enables the first and second interrupt ~~modes~~.

25. (canceled) ~~A board support package comprising:~~
~~an instruction that disables first and second interrupt types, disposed in a first vector address of an interrupt vector table executable upon receipt of an interrupt of a first type;~~
~~an other instruction that branches to a common interrupt dispatcher, disposed at a second vector address of the interrupt vector table executable upon receipt of an interrupt of a second type;~~
~~the common interrupt dispatcher having an interrupt routine that checks a mode identifier to determine whether a received interrupt was of the first or second type, processes the interrupt, and then re-enables the first and second interrupt modes.~~

26. (currently amended) An article of manufacture comprising:

a computer usable medium having a computer readable program code embodied therein, said computer usable medium having:

computer readable program code for providing a common interrupt dispatcher for a processor having:

(a) a first interrupt mode and a second interrupt mode to respectively accept interrupt requests of first and second types, the second interrupt mode having a higher priority than the first interrupt mode, both first and second interrupt modes being disableable to selectively reject interrupt requests of the first and second interrupt types;

(b) a mode status indicator to indicate the current mode of the processor;

(c) ~~an interrupt vector table having~~ a first vector address operatively associated with~~executable upon~~ receipt of ~~an~~ interrupt request<u>s</u> of the first type, and a second vector address operatively associated with~~executable upon~~ receipt of ~~an~~ interrupt request<u>s</u> of the second type, the first vector address operationally preceding the second vector address in the vector table;

computer readable program code for inserting an instruction ~~into~~<u>at</u> the first vector address that disables the first and second interrupt modes;

computer readable program code for inserting an other instruction at the second vector address, that branches to the common interrupt dispatcher;

computer readable program code for providing the common interrupt dispatcher with an interrupt routine that checks the mode identifier to determine whether a received interrupt call was a first interrupt type or a second interrupt type, processes the interrupt, and then re-enables the first and second interrupt modes.

27. (original) The article of manufacture of claim 26, further comprising:

computer readable program code for providing the interrupt routine with an instruction that branches first interrupts to a first interrupt handler, and branches second interrupts to a second interrupt handler.

28. (original) The article of manufacture of claim 27, wherein the first interrupt is an IRQ interrupt, and the second interrupt is an FIQ interrupt.

29. (canceled) ~~Computer readable program code for merging tiered interrupts into a unified interrupt handling system for an operating system of an embeddable processor, said computer readable program code comprising:~~

~~computer readable program code for providing a common interrupt dispatcher for a processor having:~~

~~a first interrupt mode and a second interrupt mode to respectively accept interrupt requests of first and second types, the second interrupt mode having a higher priority than the first interrupt mode, both~~

~~first and second interrupt modes being disableable to selectively reject interrupt requests of the first and second interrupt types;~~

~~a mode status indicator to indicate the current mode of the processor;~~

~~an interrupt vector table having a first vector address executable upon receipt of an interrupt request of the first type, and a second vector address executable upon receipt of an interrupt request of the second type, the first vector address preceding the second vector address in the vector table;~~

~~computer readable program code for inserting an instruction into~~at ~~the first vector address that disables the first and second interrupt modes;~~

~~computer readable program code for inserting an other instruction at the second vector address, that branches to the common interrupt dispatcher;~~

~~computer readable program code for providing the common interrupt dispatcher with an interrupt routine that checks the mode identifier to determine whether a received interrupt call was a first interrupt type or a second interrupt type, processes the interrupt, and then re-enables the first and second interrupt modes.~~

30. (canceled) ~~The computer readable program code of claim 29, further comprising:~~

~~computer readable program code for providing the interrupt routine with an instruction that branches first interrupts to a first interrupt handler, and branches second interrupts to a second interrupt handler.~~

31. (canceled) ~~The computer readable program code of claim 30, wherein the first interrupt is an IRQ interrupt, and the second interrupt is an FIQ interrupt.~~

32. (currently amended) A method of providing a unified interrupt handling system for an embeddable processor having multiple interrupt types, said method comprising the steps of:

(a) providing a processor having a first interrupt mode for accepting interrupt requests of a first type at a first type vector address, and a second interrupt mode for accepting interrupt requests of a second type at a second type vector address, the second interrupt mode having a higher priority than the first interrupt mode, both first and second interrupt modes being selectively disableable to selectively reject interrupt requests of the first and second interrupt types;

(b) providing a mode status indicator to indicate the current mode of the processor;

(c) the processor being configured so that ~~providing an interrupt vector table having a first vector address executable upon receipt of an interrupt~~

~~request of the first type, and a second vector address executable upon receipt of an interrupt request of the second type,~~ the first type vector address operationally precedes~~preceding~~ the second vector address~~ in the vector table~~;

(d) the processor being adapted to execute at least one instruction at the first type and second type vector addresses~~in the interrupt vector table~~ without interruption, upon receipt of an interrupt request;

(e) providing a common interrupt dispatcher;

(f) inserting an instruction ~~into~~at the first vector address that disables the second interrupt type~~mode~~;

(g) inserting at the second vector address, an other instruction that branches to the common interrupt dispatcher;

(h) providing the common interrupt dispatcher with an interrupt routine that processes the interrupt, and then re-enables the second interrupt types~~modes~~.